1 . Write a program to print numbers from 1 to 100.

Input : get the input for the n the number.

Process : using for loop print the series of the numbers.

Output : based on the condition the output is printed.

#include <stdio.h>

void main()

{

for (int i = 1; i <= 100; i++)

{

printf("%d\n", i);

}

}

Sample output:

1

2

3

4

5

6

7

8

9

10

11

12

2. Write a program to print even numbers from 1 to 50.

|  |  |
| --- | --- |
| **Input:** | The range 1 to 50 is predefined. |
| **Process:** | Loop through numbers from 1 to 50. For each number, check if it is divisible by 2 (i.e., even). |
| **Output:** | Even numbers from 1 to 50 (i.e., 2, 4, 6, ..., 50) |

#include <stdio.h>

void main()

{

for ( i = 1; i <= 50; i++)

{

if (i % 2 == 0)

{

printf("%d\n", i);

}

}

}

(Sample output :)

2

4

6

8

10

12

14

16

18

20

22

3. Write a program to find the factorial of a number.

|  |  |
| --- | --- |
| **Input:** | A single integer number n (usually ≥ 0) |
| **Process** | :Calculate factorial by multiplying all integers from 1 to n (i.e., 1 \* 2 \* 3 \* ... \* n) |
| **Output:** | The factorial value of n |

#include <stdio.h>

void main()

{

int n=3, i, fact= 1;

for(i=1;i<=n;i++)

{

fact=fact\*i;

}

printf("%d",fact);

}

sample output:

[ Enter a non-negative integer: 5

Factorial of 5 = 120 ]

4. Write a program to calculate the sum of digits of a number.

|  |  |
| --- | --- |
| **Input** | :A single integer number n (can be positive or negative) |

|  |  |
| --- | --- |
| **Process:** | Extract each digit from the number and add it to a sum. If negative, consider its absolute value. |

|  |  |
| --- | --- |
| **Output** | :The sum of the digits of the given number |

#include <stdio.h>

void main() {

int n, temp, sum = 0, digit;

printf("Enter an integer: ");

scanf("%d", &n);

if (n < 0)

{

n = -n;

}

temp = n;

while (temp > 0)

{

digit = temp % 10;

sum = sum + digit;

temp = temp / 10;

}

printf("Sum of digits = %d\n", sum);

}

Sample out:

[ Enter an integer: 1234

Sum of digits = 10 ]

5. Write a program to reverse a number.

|  |  |
| --- | --- |
| **Input** | An integer number n (positive or negative) |
| **Process** | Extract digits using modulo (%) and build the reversed number by multiplying and adding |
| **Output** | The reversed number |

#include <stdio.h>

void main() {

int n, reversed = 0, remainder;

printf("Enter an integer: ");

scanf("%d", &n);

while (n != 0)

{

remainder = n % 10;

reversed = reversed \* 10 + remainder;

n = n / 10;

}

printf("Reversed number = %d\n", reversed);

}

Sample output:

Enter an integer: 1234

Reversed number = 4321

6. Write a program to check whether a number is a palindrome.

|  |  |
| --- | --- |
| **Input** | An integer number n |
| **Process** | Reverse the number and compare it to the original number |
| **Output** | indicating whether the number is a palindrome or not |

#include <stdio.h>

void main()

{

int n, original, reversed = 0, remainder;

printf("Enter an integer: ");

scanf("%d", &n);

original = n;

while (n != 0)

{

remainder = n % 10;

reversed = reversed \* 10 + remainder;

n = n / 10;

}

if (original == reversed)

{

printf("%d is a palindrome.\n", original);

}

else

{

printf("%d is not a palindrome.\n", original);

}

}

Sample output:

Enter an integer: 121

121 is a palindrome

7. Write a program to print multiplication table of a number.

|  |  |
| --- | --- |
| **Input** | A single integer number n |
| **Process** | Multiply n by numbers 1 through 10 and print each result (n \* i) |
| **Output** | The multiplication table of the number from 1 to 10 |

#include <stdio.h>

void main() {

int n, i;

printf("Enter a number to print its multiplication table: ");

scanf("%d", &n);

printf("Multiplication table of %d:\n", n);

for (i = 1; i <= 10; i++)

{

printf("%d x %d = %d\n", n, i, n \* i);

}

}

Sample output:

Enter a number to print its multiplication table: 5

Multiplication table of 5:

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

8. Write a program to count the number of digits in a number.

|  |  |
| --- | --- |
| **Input** | An integer number n |
|  |  |

|  |  |
| --- | --- |
| **Process** | Repeatedly divide the number by 10 until it becomes 0, counting the steps |

|  |  |
| --- | --- |
| **Output** | The count of digits in the number displayed on the screen |

#include <stdio.h>

Void main() {

int n, count = 0;

printf("Enter an integer: ");

scanf("%d", &n);

if (n < 0) {

n = -n;

}

if (n == 0)

{

count = 1;

}

else

{

while (n > 0) {

n = n / 10;

count++;

}

}

printf("Number of digits = %d\n", count);

}

Sample output:

Enter an integer: 12345

Number of digits = 5

9. Write a program to print the Fibonacci series up to n terms.

|  |  |
| --- | --- |
| **Input** | An integer n (number of terms) |

|  |  |
| --- | --- |
| **Process** | Generate Fibonacci numbers starting from 0 and 1 up to n terms |

|  |  |
| --- | --- |
| **Output** | The Fibonacci series up to n terms |

#include <stdio.h>

void main() {

int n, i;

int t1 = 0, t2 = 1, nextTerm;

printf("Enter the number of terms: ");

scanf("%d", &n);

printf("Fibonacci Series: ");

for (i = 1; i <= n; ++i)

{ printf("%d", t1);

if (i != n)

{

printf(", ");

}

nextTerm = t1 + t2;

t1 = t2;

t2 = nextTerm;

}

printf("\n");

}

Sample output:

Enter the number of terms: 7

Fibonacci Series: 0, 1, 1, 2, 3, 5, 8

10. Write a program to calculate the sum of the first n natural numbers.

|  |  |
| --- | --- |
| **Input** | An integer n (number of natural numbers) entered by the user |

|  |  |
| --- | --- |
| **Process** | Calculate sum using formula sum = n \* (n + 1) / 2 or loop |

|  |  |
| --- | --- |
| **Output** | The sum of the first n natural numbers displayed on screen |

#include <stdio.h>

void main()

{

int n, sum;

printf("Enter the value of n: ");

scanf("%d", &n);

sum = n \* (n + 1) / 2;

printf("Sum of first %d natural numbers = %d\n", n, sum);

}

Sample output:

Enter the value of n: 10

Sum of first 10 natural numbers = 55